

# A light metric spanner

Lee-Ad Gottlieb

Department of Computer Science and Mathematics  
Ariel University\*

May 15, 2015

## Abstract

It has long been known that  $d$ -dimensional Euclidean point sets admit  $(1+\varepsilon)$ -stretch spanners with lightness  $W_E = \varepsilon^{-O(d)}$ , that is total edge weight at most  $W_E$  times the weight of the minimum spanning tree of the set [DHN93]. Whether or not a similar result holds for metric spaces with low doubling dimension has remained an important open problem, and has resisted numerous attempts at resolution. In this paper, we resolve the question in the affirmative, and show that doubling spaces admit  $(1+\varepsilon)$ -stretch spanners with lightness  $W_D = (\text{ddim}/\varepsilon)^{O(\text{ddim})}$ .

Important in its own right, our result also implies a much faster polynomial-time approximation scheme for the traveling salesman problem in doubling metric spaces, improving upon the bound presented in [BGK12].

---

\*Ariel, Israel. Email: [leead@ariel.ac.il](mailto:leead@ariel.ac.il)

# 1 Introduction

Let  $G = (V_G, E_G)$  be a metric graph, where vertices  $V_G$  represent points of some metric set  $S$ , while the edge weights of  $E_G$  correspond to inter-point distances in  $S$ . A graph  $R = (V_R, E_R)$  is a  $(1 + \epsilon)$ -stretch *spanner* of graph  $G = (V_G, E_G)$  if  $R$  is a subgraph of  $G$  (specifically,  $V_R = V_G$  and  $E_R \subset E_G$ ), and also  $d_R(u, v) \leq (1 + \epsilon)d_G(u, v)$  for all  $u, v \in G$ . Here,  $d_G(u, v)$  and  $d_R(u, v)$  denote the shortest path distance between  $u$  and  $v$  in the graphs  $G$  and  $R$ , respectively.

Low-stretch spanners have been the subject of intensive and broad-range study over the past three decades. Research has focused on minimizing such properties as construction time, vertex degree, graph diameter, and total edge weight – as well as possible trade-offs between these quantities – in various settings such as planar graphs or Euclidean spaces [Vai91, Sal91, Soa94, AMS94, CK95, AS97, DN97, AMS99, ADM<sup>+</sup>95, DNS95, GLN02, BGM04, DES08, ES13, DHN93, Sol11]. Of particular interest is a remarkable result of the nineties, that  $d$ -dimensional Euclidean spaces admit  $(1 + \epsilon)$ -stretch spanners with *lightness*  $W_E = \epsilon^{-O(d)}$ , meaning that the total spanner weight is at most a factor  $W_E$  times the weight of the minimum spanning tree (MST) of the set [DHN93].

The work of Gao et al. [GGN06] was the first to consider low-stretch spanners in metric spaces of low doubling dimension – a strictly more general setting than Euclidean space. This immediately spawned a long and fruitful line of work, showing that results comparable to those of Euclidean space can be obtained for these spaces as well, for construction time, degree and diameter. [CG06, GR08b, GR08a, Smi09, GKK13, CLNS15, Sol14]. However, the proof of lightness for Euclidean spanners relied heavily on the properties specific to that space (in particular, the *leapfrog property*) and so its analysis does not carry over to doubling spaces. In fact, the question of light low-stretch spanners for doubling metric spaces has resisted multiple attempts at resolution, and the problem of proving (or disproving) their existence has remained a central open problem in the study of spanners. The best lightness bound known for spanners in these metrics was  $\Omega(\log n)$  [Smi09, ES13]. Our contribution is in proving the following theorem:

**Theorem 1.1.** *Given a graph  $G$  representing a metric  $S$ , there exists a  $(1 + \epsilon)$ -stretch spanner  $R$  of  $G$  with weight  $W_D \cdot w(\text{MST}(G))$  where  $W_D = (\text{ddim}/\epsilon)^{O(\text{ddim})}$  and  $0 < \epsilon < \frac{1}{2}$ . Spanner  $R$  can be constructed in time  $(\text{ddim}/\epsilon)^{O(\text{ddim})} n \log^2 n$ .*

Hence we resolve the question in the affirmative. We first prove that graphs which admit spanning trees that are everywhere sparse also admit light low-stretch spanners (Theorem 3.4 in Section 3). We then show that doubling spaces can be decomposed into sets that are everywhere sparse, and that the light spanners on these sparse sets can be joined together into a light low-stretch spanner for the original set (Section 4).

**Related work.** Light spanners are known for only a handful of settings. These include planar graphs [ADD<sup>+</sup>93, ACC<sup>+</sup>96], unit disk graphs [KPX08], graphs which are snowflakes of metrics [GS14], and graphs of bounded pathwidth [GH12] and bounded genus [DHM10].

**Application to traveling salesman.** A polynomial-time approximation scheme (PTAS) for the Euclidean traveling salesman problem (TSP) was presented by Arora [Aro98] (for all  $d$ -dimensional space) and Mitchell [Mit99] (for the Euclidean plane). These celebrated results were further improved upon by Rao and Smith [RS98], who noted that the solution tour may be restricted to the edges of a light low-weight spanner for the set. They used this observation to improve the runtime of the Euclidean PTAS.

Building on these frameworks, Bartal et al. [BGK12] showed that in *metric* spaces a  $(1 + \varepsilon)$ -approximate tour can be computed in time roughly  $n^{2^{O(\text{ddim})}}$ . They also noted that whatever the minimum value of  $W_D$  may be, a tour can be computed in time  $2^{(W_D 2^{\text{ddim}(S)/\varepsilon})^{O(\text{ddim})} \log^2 \log n} n$ . Immediate from the work of Bartal and Gottlieb [BG13] is a better runtime of  $2^{(W_D 2^{\text{ddim}(S)/\varepsilon})^{O(\text{ddim})}} n$ , plus the time required to compute the spanner. As a consequence of Theorem 1.1, we conclude that doubling spaces admit a PTAS with runtime  $2^{(\text{ddim}(S)/\varepsilon)^{O(\text{ddim}^2)}} n + (\text{ddim}/\varepsilon)^{O(\text{ddim})} n \log^2 n$ . This is a dramatic improvement over what was previously known.<sup>1</sup>

## 1.1 Preliminaries and notation

**Graph properties.** Let the weight of an edge  $e$  ( $w(e)$ ) be its length. Let the weight of an edge set  $E$  be the sum of the weights of its edges,  $w(E) = \sum_{e \in E} w(e)$ . Similarly, the weight of any graph  $G = (V_G, E_G)$  is the weight of its edge-set,  $w(G) = w(E_G)$ . It follows that the weight of a path is its path length. For a partial graph  $R = (V_R, E_R) \subset G$  (meaning  $V_R \subset V_G$ ,  $E_R \subset E_G$ ),  $\text{diam}_G(R)$  is the diameter of  $R$  under the distance function  $d_G$ ; it is the maximum distance in  $G$  between a pair of vertices also in  $R$ . Then  $\text{diam}_R(R)$  is just the length of the longest path in  $R$ .

Let  $B(u, r) \subset V_G$  refer to the vertices of  $V_G$  contained in the closed ball centered at  $u \in V_G$  with radius  $r$ .  $B^*(u, r) \subset E_G$  is the edge set of the complete graph on  $B(u, r)$ . A graph  $G$  is said to be  $s$ -sparse if for every radius  $r$  and vertex  $v \in V$ , the weight of  $B^*(v, r) \cap E_G$  is at most  $sr$ . Let  $A(u, r_1, r_2)$  be the annulus that includes all points at distance from  $u$  in the range  $[r_1, r_2]$ , and  $A^*(u, r_1, r_2)$  the edge-set on the complete graph on  $A(u, r_1, r_2)$ . Points within distance  $r_1$  of  $u$  are within the hollow of the annulus, and not part of it.

**Doubling dimension.** For a point set  $S$ , let  $\lambda = \lambda(S)$  be the smallest number such that every ball in  $S$  can be covered by  $\lambda$  balls of half the radius, where all balls are centered at points of  $S$ . Then  $\lambda$  is the *doubling constant* of  $S$ , and the *doubling dimension* of  $S$  is  $\text{ddim} = \text{ddim}(S) = \log_2 \lambda$ . The following lemma states the well-known packing property of doubling spaces (see for example [KL04]).

**Lemma 1.2.** *If  $S$  is a metric space and  $C \subseteq S$  has minimum inter-point distance  $b$ , then  $|C| = \left(\frac{2 \text{rad}(S)}{b}\right)^{O(\text{ddim})}$ .*

**Point hierarchies.** Similar to what was described in [GGN06, KL04], a subset of points  $X \subseteq Y$  is an  $r$ -net of  $Y$  if it satisfies the following properties:

- (i) Packing: For every  $x, y \in X$ ,  $d(x, y) \geq r$ .
- (ii) Covering: Every point  $y \in Y$  is strictly within distance  $r$  of some point  $x \in X$ :  $d(x, y) < r$ .

The previous conditions require that the points of  $X$  be spaced out, yet nevertheless cover all points of  $Y$ . A point in  $X$  covering a point in  $Y$  is called a *parent* of the covered point; this definition allows for a point to have multiple parents, but we shall assign to the point a single parent arbitrarily.

---

<sup>1</sup>The result attributed to [BGK12] is found in Theorem 4.1 of the first arxiv version. The focus of [BG13] was a *linear-time* approximation scheme for Euclidean spaces under the word-RAM computational model, but aside from the time required to construct a hierarchy and spanner, as well as a MST for small areas, the entire construction holds for metric spaces. A theorem to this effect will be appended to a future version of [BG13], and is beyond the scope of this paper.

Two net-points  $x, y$  of some  $r$ -net are called  $c$ -neighbors if  $d(x, y) < cr$ . By the packing property of doubling spaces, a net-point  $x$  can have  $c^{O(\text{ddim})}$   $c$ -neighbors.

A hierarchy for a point set  $S$  is composed of nested levels of  $r$ -nets, where each level  $i$  of the hierarchy is a  $2^i$ -net of the level  $i - 1$  beneath it. Note that the distance from an  $i$ -level point to all its descendants is less than  $\sum_{j=0}^{\infty} 2^{i-j} = 2 \cdot 2^i$ . By scaling, we will assume throughout this paper that the minimum inter-point distance in all sets is at least 1. Then we have that top level  $H := \lceil \log_2 \text{diam}(S) \rceil$  contains a single point, and the 0-level contains all of  $S$ . (Below, we will find it convenient to allow for levels  $i < 0$ , and each of these levels contains all of  $S$  as well. The bottom level is  $L$ .) A full hierarchy for  $S$  constructed in time  $2^{O(\text{ddim})}n \log \Delta$ , where  $\Delta$  is the *aspect ratio* of  $S$ , the ratio between the largest and smallest inter-point distance in  $S$  [KL04]. A hierarchy can also be constructed in time  $2^{O(\text{ddim})}n \log n$ , and in this case some isolated net-points are represented implicitly [HM06, CG06]. We can also augment the hierarchy with  $c$ -neighbor lists for all hierarchical levels and points: It is known that there are only  $c^{O(\text{ddim})}n$  neighbor pairs, and that given the hierarchy they may all be discovered in  $c^{O(\text{ddim})}n$  time and space.

An  $r$ -semi-net is a net that satisfies the covering property but not the packing property. A *semi-hierarchy* is composed of levels of nested semi-nets.

**Net-Respecting spanning trees.** We will say that a spanning tree  $T$  is *net respecting (NR)* relative to a given hierarchy if for every edge  $e$  in  $T$ , both of its endpoints are  $i$ -level net-points for  $i$  satisfying  $12 \cdot 2^i \leq w(e) < 24 \cdot 2^{i+1}$ . (Recall from above that we may assume the existence of levels below  $i = 0$ .) The following lemma is adapted from Lemma 1.6 in [GKK13].

**Lemma 1.3.** *Every spanning tree  $T$  can be converted into a net-respecting spanning tree  $T'$  that visits all points visited by  $T$ , such that  $w(T') \leq 5w(T)$ .*

*Proof.* For every edge  $e = (x, y)$  in  $T$  do the following. Let  $x', y'$  be  $i$ -level net-point ancestors of  $x, y$  respectively, for the lowest  $i$  satisfying  $12 \cdot 2^i \leq d(x', y') < 24 \cdot 2^{i+1}$ . Replace edge  $e$  with long edge  $e' = (x', y')$ , and also add the short edges  $(x, x')$  and  $(y, y')$ . The short edges have weight at most  $2 \cdot 2^i$ , while by the triangle inequality  $w(e) \geq w(e') - 4 \cdot 2^i \geq 8 \cdot 2^i$ . While edge  $e'$  is net-respecting, edges  $(x, x')$  and  $(y, y')$  may not be. The short edges are themselves replaced by the procedure above. This leads to a series of edge replacements, and if the replacement adds an edge already in the set, we will remove the duplicate. The total weight of all short edges added by the recursive procedure is a geometric series summing to less than  $8 \cdot 2^i \leq w(e')$ . As the ratio of the long edge to the original edge is at most  $\frac{3}{2}$ , the total weight of all edges is  $2w(e')(1 + \frac{3}{2}) = 5w(e')$ .  $\square$

Let  $\text{MST}^{NR}$  denote the net-respecting minimum spanning tree. The following lemma related a local minimum spanning tree to a global net-respecting minimum spanning tree. It is immediate from Lemma 1.11 in [GKK13], which made a similar claim concerning TSP tours.

**Lemma 1.4.** *Let  $S$  be a point set equipped with a hierarchy, and  $S' \subset S$ . Let  $u$  be any point of  $S$ , and  $r > 0$  any value. Then*

- (i)  $w(\text{MST}^{NR}(S) \cap B^*(u, r)) \leq 14(w(\text{MST}(B(u, r))))$
- (ii)  $w(\text{MST}(B(u, r))) \leq 2w(\text{MST}^{NR}(S) \cap B^*(u, 4r)) + 2^{O(\text{ddim})}r$ .

We recall that the removal of points from a set can increase the weight of its minimum spanning tree by at most a factor of 2. Let  $S'$  consist of  $B(u, 2^i)$  along with all hierarchical ancestors of these points up to level  $i$ . Then  $S' \subset B(u, 3 \cdot 2^i)$ . We conclude that

$$\frac{1}{7}w(\text{MST}^{NR}(S') \cap B^*(u, r)) \leq w(\text{MST}(S')) \leq 4w(\text{MST}^{NR}(S) \cap B^*(u, 12r)) + 2^{O(\text{ddim})}r. \quad (1)$$

Finally, we note that for any  $S$ ,  $w(\text{MST}(S)) \leq |S| \text{diam}(S)$ . Tighter bounds are in fact known for doubling spaces [Aro98, Tal04, Smi10], but are not necessary for our results.

## 2 Background: Hierarchical spanners

Before presenting our results, we review the basic hierarchical spanner introduced by Gao et al. [GGN06], along with some small modifications suited for our purposes. This spanner provides a theoretical framework for our approach, although our ultimate construction will be significantly more involved. The results presented in this section are mostly well-known.

Throughout this paper, we will make the simplifying assumption that  $G$  is a complete graph. A *complete hierarchical spanner*  $R$  for a graph  $G$  on  $S$  is derived from the hierarchy of  $S$ . For some constant  $c$ , for each levels  $i$  we add an edge between every pair of  $i$ -level net-points with inter-point distance at most  $c2^i$ . More precisely, we add to  $E_R$  an edge between the two vertices in  $V_R$  that represent these points. Recall that the top level of the hierarchy is  $H$ , and the minimum inter-point distance in  $S$  is 1. For this construction we will assume that the bottom level of the hierarchy is  $L : \lfloor \log(1/c) \rfloor$ . Let  $M = H - L + 1$  be the number of levels in the hierarchy for  $S$ . To construct the complete hierarchical spanner, it suffices to construct a hierarchy for  $S$  and compute all relevant  $c$ -neighbors, in total time  $2^{O(\text{ddim})}n \log n + c^{O(\text{ddim})}n$  [GR08b, GR08a].

**Lemma 2.1.** *Any spanner  $R$  for graph  $G$  on  $S$  which satisfies that all  $i$ -level net-point  $c$ -neighbor pairs have 0-stretch (for all  $i$ ), is a  $(1 + \frac{32}{c})$ -stretch spanner for all of  $G$ , when  $c \geq 24$ .*

*Proof.* Consider any two net-points  $x, y$  in the bottom level  $L$  of the hierarchy of  $S$ . Let  $x', y'$  be their respective  $i$ -level ancestors, where  $i$  is the minimum value for which  $x', y'$  are  $c$ -neighbors. We have  $d_G(x', y') \leq c2^i$ . To derive a lower-bound on  $d_G(x', y')$ , let  $x'', y''$  be the  $(i-1)$ -level ancestors, and by assumption  $d_G(x'', y'') > c2^{i-1}$ . Applying the triangle inequality, we have  $d_G(x', y') \geq -d_G(x', x'') + d_G(x'', y'') - d_G(y'', y') > c2^{i-1} - 2 \cdot 2^i = (c-4)2^{i-1}$ . Since  $x', y'$  are  $c$ -neighbors,  $d_R(x', y') = d_G(x', y')$ . Applying the triangle inequality we have  $\frac{d_R(x, y)}{d_G(x, y)} \leq \frac{d_R(x, x') + d_R(x', y') + d_R(y', y)}{-d_G(x, x') + d_G(x', y') - d_G(y', y)} < \frac{d_G(x, y') + 2 \cdot 2^{i+1}}{d_G(x', y') - 2 \cdot 2^{i+1}} < \frac{(c-4)2^{i-1} + 2^{i+2}}{(c-4)2^{i-1} - 2^{i+2}} = \frac{c+4}{c-12} = 1 + \frac{16}{c-12} \leq 1 + \frac{32}{c}$ .  $\square$

The stretch of this spanner is arbitrarily low, but it has poor lightness bounds: In the trivial case where the  $n$  points of  $S$  reside on the line at intervals of distance 1, it is easy to see that  $w(E_R) = \Theta(n \log n) = \Theta(\log n \cdot w(\text{MST}(S)))$ . We may consider being more stingy with added edges by using the greedy algorithm of [DN97] (see also [GLN02]), and we call this construction the *greedy hierarchical spanner*. We build the spanner as before, but first sort all  $c^{O(\text{ddim})}n$  edges in increasing order. We consider each edge in turn, and before adding an edge between  $c$ -neighbors, we check to see if the stretch between them on the current partial spanner exceeds some threshold – for example, it is at most  $1 + b$  for some constant  $0 < b \leq 1$  – and only add the edge if the condition is met. In Lemma 2.2 below, we show that the greedy hierarchical spanner has  $1 + O(\frac{1}{c} + b)$  stretch.

**Lemma 2.2.** *A spanner  $R$  for graph  $G$  on  $S$ , for which all  $i$ -level net-point  $c$ -neighbor pairs have  $(1 + b)$ -stretch (for all  $i$ ) for any constant  $b > 0$ , is a  $(1 + \frac{32}{c} + 6b)$ -stretch spanner for all of  $G$ , when  $c \geq 24$ .*

*Proof.* The proof is similar to the proof of Lemma 2.1. Consider any two net-points  $x, y$  in the bottom level  $L$  of the hierarchy of  $S$ . Let  $x', y'$  be their respective  $i$ -level ancestors, where  $i$  is the

minimum value for which  $x', y'$  are  $c$ -neighbors:  $d_G(x', y') \leq c2^i$ . As above,  $d_G(x', y') > (c-4)2^{i-1}$ . Now noting that by construction  $d_R(x', y') \leq (1+b)d_G(x', y')$ , and applying the triangle inequality, we have  $\frac{d_R(x, y)}{d_G(x, y)} \leq \frac{d_R(x, x') + d_R(x', y') + d_R(y', y)}{-d_G(x, x') + d_G(x', y') - d_G(y', y)} < \frac{(1+b)d_G(x', y') + 2 \cdot 2^{i+1}}{d_G(x', y') - 2 \cdot 2^{i+1}} < \frac{(1+b)(c-4)2^{i-1} + 2^{i+2}}{(c-4)2^{i-1} - 2^{i+2}} = \frac{c+4}{c-12} + b \frac{c-4}{c-12} < 1 + \frac{32}{c} + 6b$ .  $\square$

In closing this section, we note that the stretch bounds of Lemmata 2.1 and 2.2 also hold for net-points of semi-hierarchies. Indeed, the proofs use only the covering property of nets, and not their packing property.

### 3 A light spanner for graphs with sparse spanning trees

In this section, we show that graphs with sparse spanning trees admit light low-stretch spanners. Note that we do not require the graph itself to be sparse, merely that it admit a sparse spanning tree. In Section 3.1, we will show that we can build a light spanner for the union of two low-stretch paths. In Section 3.2, we will build on this result to build a light spanner for all graphs with sparse spanning trees.

#### 3.1 A light spanner for pairs of close low-stretch paths

In this section we show that given a close pair  $P, Q \subset G$  of low-stretch paths, we can compute a light low-stretch spanner for their union. We first need to define nets, hierarchies and bipartite spanners for paths.

**Path nets and hierarchies.** Let  $P \subset G$  be a path consisting of some vertices and edges in  $G$ . An  $r$ -path-net for  $P$  is an  $r$ -net for  $P$  under the path distance function  $d_P$ . A *path-hierarchy* for  $P$  is a full hierarchy under  $d_P$ , with net points in levels  $i = \lceil \log_2 w(P) \rceil$  and lower.

It is easy to see that an  $r$ -path-net of  $P$  is an  $r$ -semi-net for the point of  $P$  under  $d_G$ : We observe that the distance function  $d_P$  is non-contractive with respect to  $d_G$ ;  $d_G(u, v) \leq d_P(u, v)$  for all  $u, v \in V$ . Hence, if a point is covered in the path-net, it is covered under  $d_G$  as well. However, the packing property may not hold, so the path-net is only a semi-net for  $P$  under  $d_G$ . It follows as well that the path-hierarchy for  $P$  is a semi-hierarchy for  $P$  under  $d_G$ .

**Bipartite path spanners.** Let  $G = (V_G, E_G)$  be a complete graph, and let  $P = (V_P, E_P), Q = (V_Q, E_Q)$  be two paths in  $G$  with stretch at most  $(1 + b_1)$  for some  $1 \leq b_1 \leq c$ . Let each path be equipped with a net-hierarchy. A *complete bipartite hierarchical spanner*  $R$  for  $P \cup Q$  contains all vertices of  $V_P \cup V_Q$  and path edges  $E_P \cup E_Q$  as well as the set of all edges  $E \subset E_G$  connecting vertices of  $V_P$  and  $V_Q$ . A *greedy bipartite hierarchical spanner*  $R$  for  $P \cup Q$  contains all vertices of  $V_P \cup V_Q$  and path edges  $E_P \cup E_Q$ , as well as an edge-set  $E' \subset E \subset E_G$ : We sort the edges of  $E$  in increasing order, and consider each edge in turn. An edge of  $E$  is added to  $E'$  only if the stretch between its endpoint on the current partial spanner is large, say more than  $(1 + b_1 + b_2)$  for some  $0 < b_2 \leq 1 - b_1$ . We show that the greedy bipartite spanner has favorable properties:

**Lemma 3.1.** *Let  $P, Q \subset G$  be a pair of  $(1 + b_1)$ -stretch paths whose distance from each other is not greater than  $c \cdot \min\{w(P), w(Q)\}$  (for  $0 \leq b_1 < 1$  and any  $c \geq 24$ ). Let  $\Delta$  be the aspect ratio of  $P \cup Q$ . A greedy bipartite hierarchical spanner  $R$  with parameters  $c$  and  $0 < b_2 \leq 1 - b_1$  for paths  $P, Q$  can be constructed in time  $O(cs(|V_P| + |V_Q|) \log(c\Delta))$ , and satisfies the following properties:*

- (i) *Stretch:  $d_R(p, q) \leq (1 + \frac{32}{c} + 6(b_1 + b_2))d_G(p, q)$  for all  $p \in P$  and  $q \in Q$ .*

- (ii) *Weight:*  $w(E') = \frac{12c^2s}{b_2} \cdot \min\{w(P), w(Q)\}$ , where  $E' = E_R - (E_P \cup E_Q)$  is the set of new spanner edges not in  $P$  or  $Q$ .

*Proof.* (i) This follows immediate from Lemma 2.2.

(ii) The proof is by a charging argument. We assume without loss of generality that  $w(P) \leq w(Q)$ . Take the maximum level  $i$  for which  $S$  has an  $i$ -level edge, and let the endpoints of the edge be  $p \in P$  and  $q \in Q$ . Consider any point  $\tilde{p} \in P$  satisfying  $d_P(p, \tilde{p}) \leq r$  for  $r = \frac{b_2}{4c} d_G(p, q)$ , and we will show that  $\tilde{p}$  cannot have *any* edge incident to  $Q$  in  $E_R$ :

First note that the proximity of  $p$  to  $\tilde{p}$  implies that  $\tilde{p}$  may be a  $j$ -level net-point only for values  $j \leq \log r$ . Now suppose by way of contraction that  $\tilde{p} \in P$  has an edge to some  $\tilde{q} \in Q$ , and then by construction in must be that  $d_G(\tilde{p}, \tilde{q}) \leq c2^j \leq cr$ . Let  $R'$  be the partial spanner before the addition of the  $i$ -level edges. Then the stretch guarantees of the paths and spanner, along with applications of the triangle inequality, imply that  $\frac{d_{R'}(p, q)}{d_G(p, q)} \leq \frac{d_{R'}(p, \tilde{p}) + d_{R'}(\tilde{p}, \tilde{q}) + d_{R'}(\tilde{q}, q)}{d_G(p, q)} \leq \frac{r + cr + (1+b_1)d_G(\tilde{q}, q)}{d_G(p, q)} \leq \frac{r + cr + (1+b_1)[d_G(p, q) + d_G(p, \tilde{p}) + d_G(\tilde{p}, \tilde{q})]}{d_G(p, q)} \leq \frac{(2+b_1)(r+cr) + (1+b_1)d_G(p, q)}{d_G(p, q)} = 1 + b_1 + \frac{(2+b_1)(r+cr)}{d_G(p, q)} < 1 + b_1 + \frac{4cr}{d_G(p, q)} = 1 + b_1 + b_2$ . This implies that in  $R'$  the stretch from  $p$  to  $q$  is not sufficient to add to  $E'$  an edge connecting them – a contradiction.

It follows that for any  $i$ -level edge  $e \in E'$  connecting  $p \in P$  to  $q \in Q$ , the entire path of  $P$  within distance  $\frac{b_2}{4c} w(e)$  of  $p$  under  $d_P$  has no lower level edges incident upon it. We charge the  $i$ -level edge to this segment of the path  $P$ .

The statement now follows by noting that  $p$  may have at most  $3cs$   $i$ -level edges incident upon it (all of which will be charged to the same path segment of  $P$ ): This is because all relevant  $i$ -level path-net points of  $Q$  within distance  $c2^i$  of  $p$ . Since the stretch in  $Q$  is less than 2, a path connecting all these points must lie fully within distance  $3c2^i$  of  $p$ , and by the sparsity guarantee its length is less than  $3cs2^i$ . There can be at most  $3cs$   $i$ -level path-net points on a path of this length.

**Runtime.** At each of  $O(\log(c\Delta))$  levels  $i$ , we must locate for each  $i$ -level path-net point  $p \in P$  its  $i$ -level path-net  $c$ -neighbors in  $Q$ . As shown above, there are most  $3cs$  such neighbors. Further, if two points  $p \in P$  and  $q \in Q$  are  $i$ -level  $c$ -neighbors, then so are their respective parents  $p' \in P$  and  $q' \in Q$ :  $d_G(p', q') \leq d_G(p, p') + d_G(p, q) + d_G(q, q') \leq 2 \cdot 2^{i+1} + c2^i = (\frac{c}{2} + 2)2^i + 1 < c2^{i+1}$ . So to compute all  $c$ -neighbors, it suffices to iterate down the hierarchy, maintaining for each  $i$ -level net-point  $p \in P$  a list of all  $c$ -neighbors in  $Q$ . The lists for the children of  $p$  can be found by considering all children of  $p$ 's  $c$ -neighbors, for a total of  $O(cs)$  candidates.  $\square$

### 3.2 Extension to graphs with sparse spanning trees

The results in the previous section apply only to pairs of low-stretch paths. We will use them to obtain similar results for all graphs with sparse spanning trees. Our plan is as follows: We first show how to *decompose* a spanning tree into paths without a stretch guarantee (Lemma 3.2). We then show how to *replace* any path with a light set of low-stretch paths (Lemma 3.3). Finally, we build the bipartite spanner of the previous section on all pairs of low-stretch paths, and show that this gives a light low-stretch spanner tree for the full set (Theorem 3.4). To do this, we will need to demonstrate that the union of all the low-stretch paths is still relatively sparse, and also that the union of the path-nets for these paths can serve as a semi-hierarchy for the entire space.

**Lemma 3.2.** *Given an  $s$ -sparse spanning tree  $T = (V_T, E_T) \subset G$  of  $n$  nodes with aspect ratio  $\Delta$ ,  $T$  may be decomposed in time  $O(s|V_T| \log \Delta)$  into a set  $\mathcal{Q}$  of paths, with the following properties:*

- (i) *Diameter:* At least one path  $P_i \in \mathcal{Q}$  has length  $\text{diam}_T(T)$ .

- (ii) *Proximity to path-net points: Every vertex in  $V_G$  is within distance  $b$  (under  $d_T$ ) of some path  $P_j \in \mathcal{Q}$  of length  $w(P_j) \geq b$ , for all values  $0 < b \leq \text{diam}_T(T)$ .*

*Proof.* The decomposition procedure removes from the spanning tree the longest path – of length  $\text{diam}(P)$  – and places the path in the collection  $\mathcal{Q}$ . That is, the edges of the path are removed from  $E_T$ , and then all vertices with no edges incident upon them are removed from  $V_T$ . After the removal of the longest path, a number of disjoint subtrees may remain, and each is decomposed recursively until  $V_T$  is empty. This completes the description of the decomposition procedure.

We note trivially that the distance under  $d_T$  from a removed path to all vertices in one of the remaining spanning trees is not greater than the graph diameter of that spanning tree.

(i) The first item follows by construction.

(ii) Take any vertex and the paths that were removed from the spanning tree containing the vertex at each iteration. The paths are necessarily in decreasing order of length. Consider the first path of length at most  $b$ . Then the diameter of the subtree that contained this path before its removal was at most  $b$ , hence the vertex is within distance  $b$  of the next path, which has length greater than  $b$ .

**Runtime.** First note trivially that longest path of a tree can be found in  $O(n)$  time. Now consider any subtree formed by the removal of a path  $P$  of length  $w(P)$ . After  $O(s)$  further decomposition steps, the longest path in any remaining subtrees cannot be longer than  $\frac{w(P)}{2}$ . This is because each decomposition step removes the longest path of the current subtree, and leaves behind it a group of subtrees all within distance  $w(P)$  of  $P$ . Then the fact that the original subtree is  $s$ -sparse implies that it may contain at most  $O(s)$  paths of length greater than  $\frac{W}{2}$ . It follows that the total runtime of the decomposition procedure is  $O(s|V_P| \log \Delta)$ .  $\square$

It remains to replace each of the paths in  $\mathcal{Q}$  with a set of low-stretch paths. Consider the following procedure (motivated by the greedy algorithm) which takes a path  $P$  and value  $c$ , and replaces  $P$  with a set of low-stretch paths  $\mathcal{P}$ . First consider the endpoint  $p, q \in P$ . If the endpoints are close together –  $d_P(p, q) \leq \frac{1}{3} \text{diam}_G(P)$  – find a point  $r$  satisfying  $d_G(p, r), d_G(q, r) \geq \frac{1}{3} \text{diam}_G(P)$ , segment  $P$  at  $r$ , and continue the procedure below separately on the two smaller paths. Otherwise, continue the procedure on  $P$  itself:

Iteratively create a path-net for  $P$ , at each iteration  $i = L, L+1, \dots$  promoting some of the  $(i-1)$ -level path-net points to also be  $i$ -level path-net points: We begin by promoting the first point in  $P$  – call this  $p$  – and proceeding down the path promote in turn every  $(i-1)$ -level point at path distance  $2^i$  or greater from the previously promoted point.

Then beginning again at  $p$ , we search down the path for the *associate* of  $p$ , the first  $i$ -level path-net point  $q$  satisfying  $d_G(p, q) \geq c2^i$ . Now, if the stretch  $\frac{d_P(p, q)}{d_G(p, q)}$  is greater than  $1 + \frac{1}{3c(c+1)s}$ , remove the partial path connecting  $p, q$ , and replace it by a directed edge. The partial path is then segmented into smaller paths at its  $(i-2)$ -level path-net points, and these small paths are all added to  $\mathcal{P}$ . If the partial path is removed, the procedure continues with point  $q$ , searching down the path for its associate. If the partial path is not removed, then the procedure takes the first  $i$ -level path-net point  $r$  following  $p$ , and searches down the path for  $r$ 's associate. Iteration  $i$  terminates upon reaching an  $i$ -level path-net point  $r$  with no associate. In this case, we locate the final  $i$ -level path-net point in  $P$  – call this  $q$  – and remove the partial path connecting  $p$  and  $q$ , replacing it by a direct edge. The removed partial path, as well as the path beyond  $q$ , are decomposed into smaller paths at their  $(i-2)$ -level path-net points, and added to  $\mathcal{P}$ . The procedure terminates upon reaching a level  $i$  for which the endpoint  $p$  has no associate.



Lemma 3.3 below shows that the final path set  $\mathcal{P}$  has favorable properties, which we will use in the construction of the spanner.

**Lemma 3.3.** *Let  $P = (V_P, E_P)$  be an  $s$ -sparse path in  $G$  with arbitrary stretch and aspect ratio  $\Delta$ . For any constant  $c \geq 24$ , the collection  $\mathcal{P}$  may be computed in time  $O(cs|V_P| \log \Delta)$ , and possesses the following properties:*

- (i) *Vertex cover:* The union of vertices in all sets  $P_i \in \mathcal{P}$  is exactly  $V_P$  ( $\cup_i V_{P_i} = V_P$ ), and  $\sum_i |V_{P_i}| \leq 3|V_P|$ .
- (ii) *Stretch:* Each path in  $\mathcal{P}$  has stretch at most  $(1 + \frac{32}{c})$ .
- (iii) *Diameter:* At least one path  $P_i \in \mathcal{P}$  has length  $\frac{\text{diam}_G(P)}{4} \geq \frac{\text{diam}_P(P)}{4s}$  or greater.
- (iv) *Proximity to path-net points:* Every vertex  $v \in V_P$  is within  $d_G$  distance  $16(c+1)b$  of some path  $P_j \in \mathcal{P}$  of weight  $w(P_j) \geq b$ , where  $b$  is any value satisfying  $0 < b \leq \frac{\text{diam}_G(P)}{4}$ .
- (v) *Path sparsity:* Each path in  $\mathcal{P}$  is  $3s$ -sparse.
- (vi) *Graph sparsify and weight:* The union of all paths  $P_i \in \mathcal{P}$  form a graph that is both  $3s(3cs+1)$ -sparse and has total weight at most  $3s(3c(c+1)s+1) \cdot w(P)$ .

*Proof.* (i) This covering immediate from the construction, and the number of vertices follows by noting that a removed path may duplicate the path endpoints, but removes at least a single vertex from the original path.

(ii) Let  $p, q$  be any pair of  $i$ -level path-net points in a path  $P' \in \mathcal{P}$ , and we show that the path stretch of  $p, q$  is less than  $1 + \frac{1}{c}$ . Then the item will follow from Lemma 2.2.

Since  $p, q$  are found on the same path, it must be that all  $(i+1)$ -level associate pairs  $r, t$  straddling the path segment from  $p$  to  $q$  have path stretch less than  $1 + \frac{1}{3c(c+1)s}$ , as otherwise the path connecting  $r$  and  $t$  would have been removed and segmented along its  $(i-1)$ -level path-net points, and  $p, q$  could not be in the same path segment in  $\mathcal{P}$ . Recall by construction the  $d_G$ -distance from  $r$  to  $t$  is in the range  $[c2^{i+1}, (c+1)2^{i+1})$ , and since all paths of  $\mathcal{P}$  are  $3s$ -sparse (item (iv) of the lemma), the  $d_{P'}$  distance from  $r$  to  $t$  is at most  $3s \cdot (c+1)2^{i+1}$ . Were the pair  $p, q$  to have path stretch at least  $1 + \frac{1}{c}$ , then  $r, t$  would have path stretch at least  $\frac{d_{P'}(r,t)}{d_G(r,t)} = \frac{d_{P'}(r,p) + d_{P'}(r,t) + d_{P'}(t,p)}{d_G(r,t)} \geq \frac{d_G(r,p) + (1 + \frac{1}{c})d_G(r,t) + d_{P'}(t,p)}{d_G(r,t)} \geq \frac{d_G(r,t) + \frac{2^i}{c}d_G(r,t)}{d_G(r,t)} \geq \frac{3s \cdot (c+1)2^{i+1} + 2^i/c}{3s \cdot (c+1)2^{i+1}} = 1 + \frac{1}{3c(c+1)s}$ , which is a contradiction.

(iii) The endpoints  $p, q \in P$  were chosen so that  $\text{diam}_P(P) \geq \frac{\text{diam}_G(P)}{3}$ . By construction, the final path remaining after all iterations retains  $p$  as an endpoint. On the other end, at each iteration  $i$  a small end-segment of the path may be segmented, but these account for less than a fraction  $\frac{1}{c} \sum_{i=0}^{\infty} 2^{-i} = \frac{2}{c} \leq \frac{1}{12}$  of the total path length. So the diameter is at least  $\frac{\text{diam}_G(P)}{3} \cdot \frac{11}{12} > \frac{\text{diam}_G(P)}{4}$ . Finally, the  $s$ -sparsity of path  $P$  directly implies that  $\text{diam}_P(P) \leq s \text{diam}_G(P)$ .

(iv) For every added edge connecting  $i$ -level path-net points  $p, q$ , the removed path was all within distance  $c2^i$  of  $p$  (under  $d_G$ ) of the new edge connecting  $p, q$ , which have length in the range  $d_G(p, q) \in [c2^i, (c+1)2^i)$ . Now the removed path may itself have had paths removed from it at earlier stages of the iteration, but all these paths must be within distance  $\sum_{j=0}^i c2^j < c2^{i+1} < 2d_G(p, q)$  of the new edge. Also note that when this path was removed, it was decomposed into smaller paths of

path-length at least  $2^{i-2} = \frac{2^{i+1}}{8}$ . Hence, the length of the edges feature jumps of at most  $8(c+1)$ , and so every vertex is within distance  $16(c+1)b$  of an edge of length at least  $b$ .

(v) Take an edge added by the procedure between points  $p, q$ . Then in the original path  $P$  there must have been a heavier path connecting  $p, q$ , and by the arguments in item (iv) this path was fully within distance  $d_G(p, q)$  of both  $p$  and  $q$ . Any ball containing the new edge must be of radius at least  $\frac{d_G(p, q)}{2}$ , and so it contains the entire (removed) old path within 3 times its radius. So the weight of the smaller ball is less than the weight of the larger ball, and the sparsity is  $3s$ .

(vi) For the claim of weight: We have already noted in item (iii) that the edges added when removing an end-segment sum to much less than the weight of the path. For the other edges, the proof proceeds by a charging argument. For an edge  $e$  added between points  $p, q$  in iteration  $i$ , charge its weight to the edges of the removed path. Each edge  $e'$  in the removed partial path  $Q$  receives charge  $w(e) \frac{w(e')}{w(R)}$ , and since  $Q$  must have weight at least  $w(e)(1 + \frac{1}{3c(c+1)s})$ , the charge to  $e'$  is at most  $\frac{w(e')}{1 + \frac{1}{3c(c+1)s}}$ . Now, if  $e'$  itself was added in an earlier iteration  $j$ , then the new charge to  $e'$  due to the addition of  $e$  is itself passed down to earlier removed paths, until all charges lie solely on edges present in the original path  $P$ . Let  $e$  be an edge of  $P$ , and it follows that the sum of charges placed on  $e$  (included the cost of  $e$  itself) is less than  $w(e) \sum_{i=0}^{\infty} (1 + \frac{1}{3c(c+1)s})^{-i} < (3c(c+1)s+1)w(e)$ .

For the claim of sparsity: It follows from the proof of items (iii) and (iv) that all edges of  $\mathcal{P}$  inside an  $r$ -radius ball charge to edges of the original path  $P$  within distance  $3r$  of the ball's center. As  $P$  is  $s$ -sparse, the total weight in the small ball is  $3s(3c(c+1)s+1)$ , and the total weight of all edges in  $\mathcal{P}$  is  $3s(3c(c+1)s+1) \cdot w(P)$ .

**Runtime.** There are  $O(\log \Delta)$  iterations, and at each one we can make a single pass on the path, promoting net-points and recording their order in a list. Then we must discover for each  $i$ -level path-net point  $p$  its  $c$ -neighbors. But these are all within path distance  $O(cs2^i)$  of  $p$ , so it suffices to inspect only the next  $O(cs)$  net-points in the list. The runtime follows.  $\square$

Joining together Lemmata 3.1, 3.2 and 3.3, we can prove that spaces with sparse spanning trees admit light low-stretch spanners:

**Theorem 3.4.** *Let  $T = (V_T, E_T)$  be an  $s$ -sparse spanning tree of a complete graph  $G$  with aspect ratio  $\Delta$ . Then in time  $(s/\varepsilon)^{O(1)} \cdot |V_T| \log \Delta$  we can construct for  $G$  a spanner  $R$  with*

(i) *Weight:*  $W_s \cdot w(T)$  for  $W_s = (s/\varepsilon)^{O(1)}$ .

(ii) *Stretch:*  $(1 + \varepsilon)$ .

*Proof.* The construction is straightforward: Decompose the spanning tree  $T$  into paths using the *decomposition* procedure of Lemma 3.2, and then replace each path with a set of low-stretch paths using the *replacement* procedure of Lemma 3.3 with parameter  $c_1 = 3 \cdot 2^7/\varepsilon$ . For every pair of paths in the new set, build the greedy bipartite spanner of Lemma 3.1 with parameters  $c_2 = \max\{16c_1 + 18, 8s\} \cdot 2^6/\varepsilon$  and  $b_2 = \varepsilon/12$ . The new graph is  $G'$ . Recall that the bipartite spanner will only add edges if the path pair is sufficiently close.

(i) We first must prove sparsity: Tree  $T$  is  $s$ -sparse, and the sparsity condition is not affected by the decomposition. Take any path  $P$  the decomposition, and  $P$  is replaced by a set of paths connecting the vertices of  $P$  with sparsity  $O(c_1^2 s)$  (Lemma 3.3(vi)). Hence, before the addition of the bipartite edges  $G'$  possessed sparsity  $s' = O(c_1^2 s^2)$ .

Returning to proof of weight, it follows from Lemma 3.3(vi) that the weight of the paths of  $G'$  before the construction of the bipartite spanners was  $3s(3c_1(c_1+1)s+1) \cdot w(T)$ . A bipartite spanner

with parameter  $c_2$  is built for each pair of paths, and we will charge the cost of the new edges to the shorter path. By sparsity, any path  $P$  is within distance  $c_2 \cdot w(P)$  of  $O(c_2 s') = O(c_1^2 c_2 s^2)$  longer paths. Charge  $P$  for the additional edges added by the bipartite spanner to each of these paths. By Lemma 3.1 this yields a charge of  $O(c_2^2 s/b_2) \cdot w(P)$  per spanner, for a total charge of  $O(c_1^2 c_2^3 s^3/b_2) \cdot w(P)$ . It follows that  $w(G') = O(c_1^2 c_2^3 s^3/b_2) \cdot w(T)$ .

(ii) By Lemma 3.3, the intra-path stretch in all paths produced by the replacement procedure is  $1 + \frac{32}{c_1}$ . By construction of the bipartite spanner, the stretch among all  $i$ -level path-net points that are  $c_2$ -neighbors is  $\max\{1 + b_2, 1 + \frac{32}{c_1}\}$ .

We require a semi-hierarchy: Below we show that each vertex  $v$  in  $G'$  is within distance  $(16c_1 + 18)2^i$  of some  $i$ -level path-net point for all levels  $i \leq \lceil \text{diam}_T(T)/8s \rceil$ . It follows that if we promote all  $i$  level path-net points (for each  $i$ ) to be  $j$ -level points for  $j = i + \log \max\{16c_1 + 18, 8s\}$ , then each vertex  $v$  is within distance at most  $2^j$  of some  $j$ -level point, and this hold for all levels  $j \leq \lceil \text{diam}_T(T) \rceil = H$ . Hence we have a valid semi-hierarchy for all points. Substituting  $j$  into the stretch bound above, we have that the the stretch among all  $j$ -level semi-hierarchy net-points that are  $\frac{c_2}{\max\{16c_1 + 18, 8s\}}$ -neighbors is  $\max\{1 + b_2, 1 + \frac{32}{c_1}\}$ . Substituting in the values of  $c_1, b_2$  chosen above, we have that the the stretch among all  $j$ -level semi-hierarchy net-points that are  $(64/\varepsilon)$ -neighbors is  $1 + \frac{\varepsilon}{12}$ . Then by Lemma 2.2, the resulting spanner has stretch  $1 + \frac{32}{64/\varepsilon} + 6\frac{\varepsilon}{12} = 1 + \varepsilon$ .

It remains only to verify the above claim, that each vertex in  $G'$  is within distance  $(16c_1 + 18)2^i$  of some (not yet promoted)  $i$ -level path-net point: After the decomposition procedure, for any value  $0 < b \leq \text{diam}_T(T)$  each vertex  $v \in V_G$  is within distance  $b$  of some vertex  $w \in P_i$  for which  $w(P_i) \geq b$  (Lemma 3.2(ii)). After the replacement procedure,  $w$  is within distance  $16(c_1 + 1)b'$  of some path of length at least  $b'$  where  $0 < b' \leq \frac{\text{diam}_P(P_i)}{4s}$ . Setting  $b' = b$ , we have that  $v$  is within distance  $(16c_1 + 17)b$  of some path of length at least  $b$ , where  $0 < b \leq \frac{\text{diam}_T(T)}{4s}$ . A path of this length possesses a path-net point at level  $i = \lceil \log(b/2) \rceil$  or greater. It follows that  $v$  is within distance  $(16c_1 + 18)2^i$  of some  $i$ -level path-net point for all levels  $i \leq \lceil \text{diam}_T(T)/8s \rceil$ .

**Runtime.** By Lemmata 3.2 and 3.3, the decomposition and replacement steps can be done in time  $O(c_2 s |V_T| \log \Delta)$ . To compute each bipartite spanners, for each path  $P$  in  $G'$  we must have all paths of length  $w(P)$  or greater within distance  $c_2 \cdot w(P)$  of  $P$ . Recall from the proof of (i) above that the sparsity of  $G'$  before the addition of bipartite spanner edges is  $s' = (s/\varepsilon)^{O(1)}$ .

After the initial decomposition step, assign a path-hierarchy to each path, and compute all  $i$ -level  $2c_2$ -neighbors among all paths and pairs of paths, for all  $i$ . This can all be done in time  $(s/\varepsilon)^{O(1)} n \log \Delta$ . (The construction and analysis is identical to that of Lemma 3.2.) For each path  $e$  created by the replacement step, we compute its distance to all  $i$ -level path-net points in the original edge within distance  $c_2 2^i$ . Since the current graph is  $s'$ -sparse and has aspect ratio  $\Delta$  this can be done in time  $(s/\varepsilon)^{O(1)} n \log \Delta$  over all paths, if we precomputed for each vertex its closest  $i$ -level path-net point for all  $i$ . For each  $i$ -level path-net point, we maintain a list of the replacement path which discovered this point, sorted in order of path length. Note that we may sort all replacement paths once, and then compute close path-net point for each one in turn.

For each path  $P$  and each  $i \leq \log(w(e))$ , we inspect the  $i$ -level path-net points within distance  $c_2 2^i$ , and all  $i$ -level  $c_2$ -neighbors of those path net points, and their listed paths of length greater than  $w(P)$ , and this suffices to discover for  $P$  all longer edges within distance  $c_2 \cdot w(P)$ . The runtime follows from the sparsity and aspect ratio of the current graph.  $\square$

## 4 A light spanner for general metric graphs

In Section 3 we showed how to construct a light spanner for spaces that have sparse spanning trees. In this section we complete the proof of Theorem 1.1 by showing how to decompose general metric graphs of low doubling dimension into graphs with sparse spanning trees. Then the light spanners for the decomposed sparse spaces can all be joined into a single light spanner for the original metric space. Our approach uses a technique developed for computing near-optimal traveling salesman tours in polynomial time [BGK12, BG13], although our setting is much less restrictive, and so the problem of finding a good decomposition is simpler.

For some graph  $G$  equipped with a hierarchy, define  $F(u, i)$  for  $i$ -level vertex  $u$  to include all points of  $B(u, i) \cap G$ , as well as all their hierarchical ancestors up to level  $i$ . The following preliminary lemma shows that we can spin off a (slightly) dense area of the graph:

**Lemma 4.1.** *Let  $G$  be a graph with aspect ratio  $\Delta$ , equipped with a hierarchy. For some fixed  $f = (\text{ddim}/\varepsilon)^{O(\text{ddim})}$ , let  $i$  be the lowest level for which there exists some  $u$  satisfying  $\text{MST}^{NR}(F(u, i)) > f2^i$ . Then  $G$  may be segmented into intersecting subgraphs  $G', D \subset G$  with the following properties, for any value  $c \geq 2$ :*

- (i) *Weight in  $G'$ :  $w(\text{MST}^{NR}(F(u, i) \cap G')) \leq \frac{f}{4}2^i$ , while  $w(\text{MST}^{NR}(G)) - w(\text{MST}^{NR}(G')) \geq \frac{f}{8}2^i$ .*
- (ii) *Sparsity of  $D$ :  $\text{MST}(D)$  is  $c^{O(\text{ddim})}f$ -sparse, while  $D$  has diameter  $O(c) \cdot 2^i$ .*
- (iii) *Neighbor proximity: For every  $c$ -neighbor point pair in the hierarchy of  $G$ , the pair is found together in  $A$  or  $B$  or in both.*

*Proof.* Subset  $D$  includes all points in  $B(u, (13 + c) \cdot 2^i)$ , along with all their net-points up to level  $i$ . Subset  $G'$  includes all points of  $G - B(u, 13 \cdot 2^i)$ , as well as all  $i$ -level (and higher level) net-points in  $B(u, 13 \cdot 2^i)$ . Below, we will also add some more points to  $A$ .

(ii) Recall that by assumption each  $(i - 1)$ -level point  $v$  satisfies  $\text{MST}^{NR}(F(v, i - 1)) \leq f2^{i-1}$ . A spanning tree for any ball of radius  $2^i \leq r \leq (13 + c) \cdot 2^i$  in  $D$  can be formed by covering that ball with the spanning trees of  $r^{O(\text{ddim})}$  sets  $F(v, i - 1)$ , and then connecting the centers of these sets at an additional cost of  $r^{O(\text{ddim})}2^i$ , for a total weight of  $r^{O(\text{ddim})}f2^i$ . By the optimality of  $\text{MST}(D)$ , its weight inside the given ball cannot be greater than this. A similar result holds for values of  $r$  less than  $2^i$ , covering with sets  $F(v, k)$  for  $k < i - 1$ .

(iii) Clearly  $c$ -neighbor pairs of levels  $i$  or higher are both found in  $G'$ . For levels  $k < i$ , if one of the points is not found in  $G'$ , then by construction it is found in  $B(u, 13 \cdot 2^i)$ , and so the second point is found in  $B(u, 13 \cdot 2^i + c2^k) \subset B(u, (13 + c) \cdot 2^i) = D$ .

(i) We first discuss the weight of  $\text{MST}^{NR}(G)$  in the area around  $u$ : A consequence of Equation (1) is that  $B^*(u, 12 \cdot 2^i)$  had intersected edges of  $\text{MST}^{NR}(G)$  weighing more than  $\frac{f}{4}2^i$ . At the same time, the larger ball  $B(u, 13 \cdot 2^i)$  admitted a minimum spanning tree of weight  $2^{O(\text{ddim})}f2^i$  (by arguments identical to the proof of item (ii) above). So by Lemma 1.4(i)  $B(u, 13 \cdot 2^i)$  intersected edges of  $\text{MST}^{NR}(G)$  of weight  $2^{O(\text{ddim})}f2^i$ .

We now add some more points to  $G'$ : Set  $j = i - a \log \text{ddim}$  for some absolute constant  $a > 2$  to be specified below. Add to  $G'$  all  $j$ -level points in  $B(u, 13 \cdot 2^i)$ . Further, let radius  $r \in [12 \cdot 2^i + 72 \cdot 2^j, 13 \cdot 2^i - 72 \cdot 2^j]$  be such that  $w(\text{MST}^{NR}(G) \cap A^*(u, r - 72 \cdot 2^j, r + 72 \cdot 2^j)) \leq \frac{1}{4}w(\text{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$  – that is, the weight of  $\text{MST}^{NR}(G)$  inside the annulus is a fraction of that inside the hollow. (Such a value  $r$  must in fact exist: There are  $\Theta(2^i/2^j) = \Theta(2^a \text{ddim})$

non-intersecting annuli. If each contained edges of  $\text{MST}^{NR}(G)$  weighing a fraction at least  $\frac{1}{4}$  of the edge-weight within their hollow, the total edge-weight within  $B(u, 13 \cdot 2^i)$  would be  $(1 + \frac{1}{4})^{2^a \text{ddim}} f 2^i$ ; for a sufficiently large choice of  $a$ , this exceed the upper-bound proved above.) Add to  $G'$  all points of the smaller annulus  $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$ , as well as all points outside outer edge of the annulus – in short, all points outside  $B(u, r - 24 \cdot 2^j)$ .

To prove the first part of item (i), note that  $F(u, i) \cap G'$  contains only points of level  $j$  or higher, and these can be connected by an MST of weight  $(2^i/2^j)^{O(\text{ddim})} 2^i = \text{ddim}^{O(2^a \text{ddim})} 2^i \ll \frac{f}{20} 2^i$ . The final inequality follows when  $f$  is chosen to be sufficiently large with respect to  $a$ . Then by Lemma 1.3, a connecting net-respecting MST has weight less than  $\frac{f}{4} 2^i$ .

To prove the second part of item (i), we will construct a net-respecting spanning graph for  $G'$  with smaller weight than  $\text{MST}^{NR}(G)$ . Recall that  $G$  and  $G'$  differ only on points within  $B(u, r - 24 \cdot 2^i)$ . We add to the spanning graph of  $G'$  all edges of  $\text{MST}^{NR}(G)$  fully outside of  $B(u, r + 24 \cdot 2^i)$  (and this is outside the smaller annulus  $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$ ). We also add to  $G'$  those edges with only a single endpoint inside  $B(u, r + 24 \cdot 2^i)$ , and these net-respecting edges must be incident on the annulus  $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$  if they are shorter than  $48 \cdot 2^j$ , or on  $j$ -level points in  $(u, r - 24 \cdot 2^j)$  if they are longer. We then claim that the points of  $G'$  inside the ball  $B(u, r + 24 \cdot 2^i)$  can be covered by a graph whose weight is lighter than that of  $\text{MST}(G) \cap B^*(u, r + 24 \cdot 2^i)$  by at least  $\frac{f}{8} 2^i$ , from which the second part of the item follows.

First note that a graph connecting all  $j$ -level points in  $B(u, 13 \cdot 2^i)$  has weight much less than  $\frac{f}{80} 2^i$  for appropriately chosen  $f$  (as above in the proof of the first part of (i)), and so by Lemma 1.3 a net-respecting MST connecting only these points has weight much less than  $\frac{f}{16} 2^i$ . Further, all points of the annulus  $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$  can be connected to some  $j$ -level point at total cost at most  $w(\text{MST}^{NR}(G) \cap A^*(u, r - 72 \cdot 2^j, r + 72 \cdot 2^j))$  (which was chosen above to be at most  $\frac{1}{4} w(\text{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$ ): The annulus cuts  $w(\text{MST}^{NR}(G))$  into disjoint paths. For each such path, if it is incident on a  $j$ -level (or higher level) point then we are done. Otherwise, consider the tail of this path, the final segment exiting the smaller annulus  $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$ . Since  $\text{MST}^{NR}(G)$  is net-respecting, and the path touches only net-points of level less than  $j$ , the tail must be of length at least  $24 \cdot 2^j$ , since it cannot jump immediately to a point outside the larger annulus. Cut off this tail, and instead add to the path a much shorter tail of length at most  $2 \cdot 2^j$  connecting to the closest  $j$ -level point. So the total cost of the spanning graph for  $G' \cap B(u, r + 24 \cdot 2^i)$  is at most  $\frac{f}{16} 2^i + \frac{1}{4} w(\text{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$ , while  $\text{MST}^{NR}(G) \cap B^*(u, r + 24 \cdot 2^i)$  is of course at least  $w(\text{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$ , a difference of  $\frac{3}{4} w(\text{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j)) - \frac{f}{16} 2^i \geq \frac{3}{4} w(\text{MST}^{NR}(G) \cap B^*(u, 12 \cdot 2^i)) - \frac{f}{16} 2^i \geq \frac{3}{4} \cdot \frac{f}{4} 2^i - \frac{f}{16} 2^i = \frac{f}{8} 2^i$ .  $\square$

By using the technique of Lemma 4.1 to repeatedly spin off relatively sparse areas of the graph, we can decompose the entire graph into relatively sparse areas, as in the following theorem:

**Theorem 4.2.** *Any graph  $G$  with aspect ratio  $\Delta$  and equipped with a hierarchy, may be segmented into a set of subgraphs  $\mathcal{D}$  ( $\cup_i D_i = G$ ) with the following properties, for any  $c \geq 2$  and for  $f$  as above:*

- (i) *Sparsity: For all  $D_i \in \mathcal{D}$ ,  $\text{MST}(D_i)$  is  $c^{O(\text{ddim})} f$ -sparse.*
- (ii) *Weight:  $\sum_i w(\text{MST}(D_i)) = c^{O(\text{ddim})} w(\text{MST}^{NR}(G))$*
- (iii) *Neighbor Proximity: For every  $c$ -neighbor point pair in the hierarchy of  $G$ , the pair is found together in at least one set  $D_i \in \mathcal{D}$ .*

(iv) *Point occurrence:* Each point of  $G$  appears in at most  $c^{O(\text{ddim})} \log \Delta$  different sets in  $\mathcal{D}$ .

The segmentation can be computed in time  $c^{O(\text{ddim})} n \log n (\log n + \log \Delta)$ .

*Proof.* The technique of Lemma 4.1 requires knowledge of  $\text{MST}^{NR}(G)$  (to compute the annulus), so we must first build this graph. We construct the full hierarchical spanner with parameter  $c$ . This can all be done in time  $c^{O(\text{ddim})} n \log n$ , and the resulting graph has only  $c^{O(\text{ddim})} n$  edges [GR08b, GR08a]. We then extract from the edges only those that are net-respecting, and build  $\text{MST}^{NR}(G)$  using this set, in total time  $c^{O(\text{ddim})} n \log n$ . We will need to maintain  $\text{MST}^{NR}(G)$  under a long series of point deletions, and this can be done in total time  $c^{O(\text{ddim})} n \log^2 n$  (see Theorem 6 in [HdLT01]).

Beginning at the lowest hierarchical level  $L$  and iterating upwards, for each  $i$ -level net-point  $u$  we compute  $F(u, i)$  and construct  $\text{MST}^{NR}(F(u, i))$ , applying the segmentation technique of Lemma 4.1 if necessary. The subset  $D$  is added to  $\mathcal{D}$ , and we iterate the procedure on subset  $G'$  until there are no more heavy neighborhoods, at which point the final subset  $G'$  is added to  $\mathcal{D}$  as well.

The sparsity of sets in  $\mathcal{D}$  and neighbor proximity follow immediately from Lemma 4.1(ii) and (iii), respectively. The bound on point occurrences follows from the packing property and height of the hierarchy, and a corollary of this bound is that  $c^{O(\text{ddim})} n \log n \log \Delta$  time suffices to compute all graphs  $\text{MST}^{NR}(F(u, i))$ . For the weight, let  $G_i$  be the graph in the  $i$ -th iteration, and  $w_i = w(\text{MST}^{NR}(G_{i+1})) - w(\text{MST}^{NR}(G_i))$ . Clearly  $\sum_i w_i \leq w(\text{MST}^{NR}(G))$ . By Lemma 4.1(i) and (ii),  $\sum_i w(\text{MST}(D_i)) = \sum_i c^{O(\text{ddim})} w_i = c^{O(\text{ddim})} w(\text{MST}^{NR}(G))$ .  $\square$

We can now complete the proof of Theorem 1.1:

*Proof.* Fix  $c = 64/\varepsilon$ . Given graph  $G$ , we build a full hierarchy for the entire space, and add to the spanner  $R$  the edges of the complete hierarchical  $(1 + \frac{\varepsilon}{12})$ -stretch spanner of Lemma 2.1 on all levels below  $i = H - \log(n^2)$ . Note that  $2^H \leq \text{diam}(S)$ , and so the longest edge added by this construction is of length  $O(1/\varepsilon) \cdot 2^{H - \log(n^2)} = O(\text{diam}(S)/\varepsilon n^2)$ . So the entire cost of all these edges is less than  $O(1/\varepsilon) \cdot \text{MST}(G)$ .<sup>2</sup> We then remove from consideration all net-points below level  $i = H - \log(n^2)$ , set  $L = i$ , and focus on the remaining  $M = H - L = O(\log n)$  levels. Note that the aspect ratio of this set is only  $\Delta = 2^{O(M)} = n^{O(1)}$ .

We apply Theorem 4.2 to decompose the remaining graph into all sparse sets in time  $(\text{ddim}/\varepsilon)^{O(\text{ddim})} n \log^2 n$ , and construct the  $(1 + \frac{\varepsilon}{12})$ -stretch spanner of Theorem 3.4 for each sparse subset. Since any point of  $G$  appears in most  $(1/\varepsilon)^{O(\text{ddim})} \log n$  sets (Theorem 4.2(iv)) the total time to construct all these spanners is  $(\text{ddim}/\varepsilon)^{O(\text{ddim})} n \log^2 n$ . Then the stretch between any pair of  $c$ -neighbors is  $(1 + \frac{\varepsilon}{12})$ , and it follows from Lemma 2.2 that the total graph stretch is  $(1 + \varepsilon)$ .

The weight guarantee follows from Theorem 4.2(ii) in conjunction with Theorem 3.4(i).  $\square$

**Open problems.** Our results suggest several avenues for further research. Can  $W_D$  be reduced to  $\varepsilon^{-O(\text{ddim})}$ , to match what is known for Euclidean space? Also, does the simple greedy hierarchical spanner alone yield a light spanner? One can also improve the runtime of our construction.

---

<sup>2</sup> This observation is due to Shay Solomon, and is similar to one made by Arora in the context of Euclidean TSP [Aro98].

## References

- [ACC<sup>+</sup>96] Srinivasa Arikati, Danny Z. Chen, L.Paul Chew, Gautam Das, Michiel Smid, and Christos D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In Josep Diaz and Maria Serna, editors, *Algorithms . ESA '96*, volume 1136 of *Lecture Notes in Computer Science*, pages 514–528. Springer Berlin Heidelberg, 1996.
- [ADD<sup>+</sup>93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9:81–100, January 1993.
- [ADM<sup>+</sup>95] Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel Smid. Euclidean spanners: Short, thin, and lanky. In *27th annual ACM symposium on Theory of computing*, pages 489–498, 1995.
- [AMS94] S. Arya, D.M. Mount, and M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 703–712, Nov 1994.
- [AMS99] S. Arya, D. M. Mount, and M. Smid. Dynamic algorithms for geometric spanners of small diameter: Randomized solutions. *Computational Geometry: Theory and Applications*, 13:91–107, 1999.
- [Aro98] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45:753–782, 1998.
- [AS97] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. *Algorithmica*, 17(1):33–54, 1997.
- [BG13] Yair Bartal and Lee-Ad Gottlieb. A linear time approximation scheme for euclidean tsp. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13*, pages 698–706, 2013.
- [BGK12] Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proceedings of the 44th symposium on Theory of Computing, STOC '12*, pages 663–672, 2012. Available at <http://arxiv.org/abs/1112.0699>.
- [BGM04] P. Bose, J. Gudmundsson, and P. Morin. Ordered theta graphs. *Computational Geometry: Theory and Applications*, 28:11–18, 2004.
- [CG06] Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *38th annual ACM symposium on Theory of computing*, pages 574–583, 2006.
- [CK95] P. B. Callahan and S.R. Kosaraju. A decomposition of multidimensional point sets with applications to  $k$ -nearest-neighbors and  $n$ -body potential fields. *J. ACM*, 42:67–90, 1995.

- [CLNS15] T.-H. Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM Journal on Computing*, 44(1):37–53, 2015.
- [DES08] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 519–528, Oct 2008.
- [DHM10] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.
- [DHN93] Gautam Das, Paul Heffernan, and Giri Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proceedings of the Ninth Annual Symposium on Computational Geometry, SCG '93*, pages 53–62, New York, NY, USA, 1993. ACM.
- [DN97] Gautam Das and Giri Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. *Int. J. Comput. Geometry Appl.*, 7(4):297–315, 1997.
- [DNS95] Gautam Das, Giri Narasimhan, and Jeffrey Salowe. A new way to weigh malnourished Euclidean graphs. In *6th annual ACM-SIAM symposium on Discrete algorithms*, pages 215–222. SIAM, 1995.
- [ES13] Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin and lanky. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, STOC '13*, pages 645–654, New York, NY, USA, 2013. ACM.
- [GGN06] Jie Gao, Leonidas J. Guibas, and An Nguyen. Deformable spanners and applications. *Comput. Geom. Theory Appl.*, 35(1), 2006.
- [GH12] Michelangelo Grigni and Hao-Hsiang Hung. Light spanners in bounded pathwidth graphs. In Branislav Rován, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 467–477. Springer Berlin Heidelberg, 2012.
- [GKK13] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate lipschitz extension. In *Similarity-Based Pattern Recognition - Second International Workshop, SIMBAD 2013, York, UK, July 3-5, 2013. Proceedings*, pages 43–58, 2013.
- [GLN02] Joachim Gudmundsson, Christos Levkopoulos, and Giri Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal on Computing*, 31:2002, 2002.
- [GR08a] Lee-Ad Gottlieb and Liam Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '08*, pages 591–600, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
- [GR08b] Lee-Ad Gottlieb and Liam Roditty. An optimal dynamic spanner for doubling metric spaces. In *Proceedings of the 16th annual European symposium on Algorithms, ESA '08*, pages 478–489. Springer-Verlag, 2008.



- [GS14] Lee-Ad Gottlieb and Shay Solomon. Light spanners for snowflake metrics. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, New York, NY, USA, 2014. ACM.
- [HdLT01] Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001.
- [HM06] Sarel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.
- [KL04] Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.
- [KPX08] Iyad A. Kanj, Ljubomir Perkovi, and Ge Xia. Computing lightweight spanners locally. In *Distributed Computing*, volume 5218 of *Lecture Notes in Computer Science*, pages 365–378. Springer Berlin Heidelberg, 2008.
- [Mit99] Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP,  $k$ -MST, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999.
- [RS98] Satish B. Rao and Warren D. Smith. Approximating geometrical graphs via “spanners” and “banyans”. In *30th annual ACM symposium on Theory of computing*, pages 540–550. ACM, 1998.
- [Sal91] J. S. Salowe. Constructing multidimensional spanner graphs. *Int. J. Comput. Geometry Appl*, 1(2):99–107, 1991.
- [Smi09] Michiel Smid. The weak gap property in metric spaces of bounded doubling dimension. In *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of his 60th Birthday*, pages 275–289. Springer-Verlag, 2009.
- [Smi10] Michiel Smid. On some combinatorial problems in metric spaces of bounded doubling dimension. Manuscript, available at <http://people.scs.carleton.ca/~michiel/research.html>, 2010.
- [Soa94] J. Soares. Approximating Euclidean distances by small degree graphs. *Discrete & Computational Geometry*, 11:213–233, 1994.
- [Sol11] Shay Solomon. An optimal-time construction of sparse euclidean spanners with tiny diameter. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 820–839. SIAM, 2011.
- [Sol14] Shay Solomon. From hierarchical partitions to hierarchical covers: Optimal fault-tolerant spanners for doubling metrics. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 363–372, New York, NY, USA, 2014. ACM.

- [Tal04] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *36th annual ACM symposium on Theory of computing*, pages 281–290. ACM, 2004.
- [Vai91] P. M. Vaidya. A sparse graph almost as good as the complete graph on points in K dimensions. *Discrete & Computational Geometry*, 6:369–381, 1991.